InstanceDiffusion: Instance-level Control for Image Generation

Xudong Wang[1,2] Trevor Darrell[2] Sai Saketh Rambhatla[1] Rohit Girdhar[1] Ishan Misra[1]
[1]GenAI, Meta [2]UC Berkeley

code: https://github.com/frank-xwang/InstanceDiffusion/

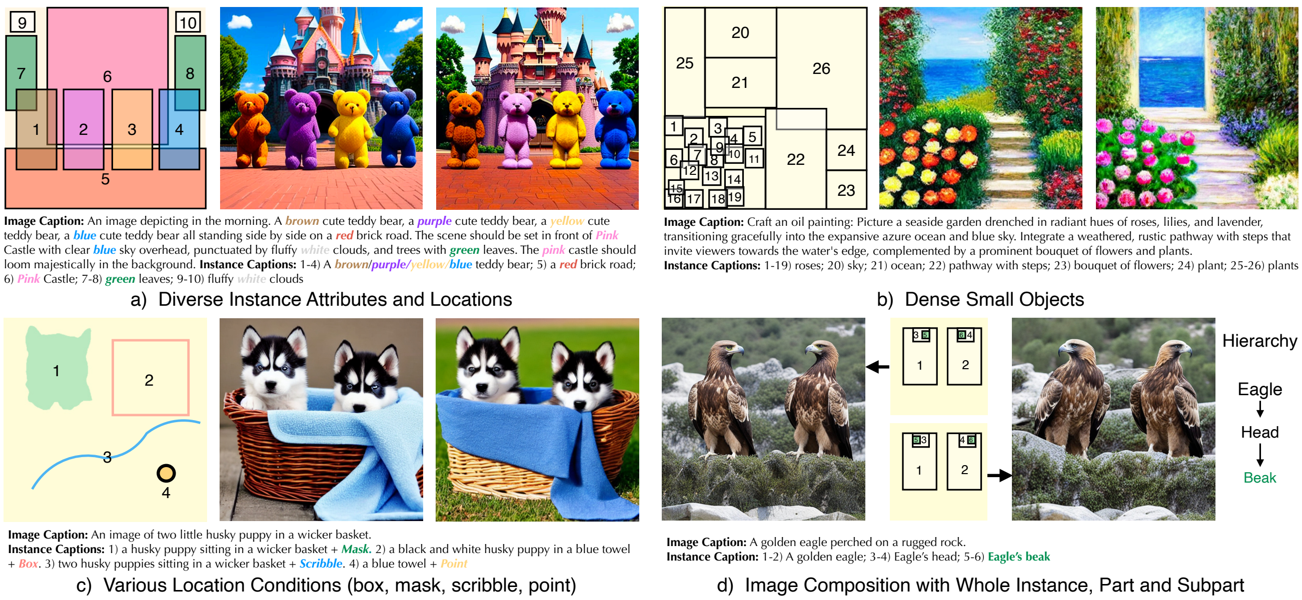

Figure 1. InstanceDiffusion's generations using instance-level text prompts and location conditions for image generation. Our model can respect: a) a variety of instances with diverse attributes (8 colors) and boxes, b) densely-packed instances (>25 objects), c) mixed location conditions (such as boxes, masks, scribbles, and points), and d) compositions with granularity spanning from entire instances to parts and subparts. The positioning of parts/subparts implicitly alters the overall pose of the object. The instance inputs and their global text prompts are displayed, with the location conditions displayed on the left image. Numbers in the box/mask/scribble/point refer to the instance id.

Abstract

Text-to-image diffusion models produce high quality images but do not offer control over individual instances in the image. We introduce InstanceDiffusion that adds precise instance-level control to text-to-image diffusion models. InstanceDiffusion supports free-form language conditions per instance and allows flexible ways to specify instance locations such as simple single points, scribbles, bounding boxes or intricate instance segmentation masks, and combinations thereof. We propose three major changes to text-to-image models that enable precise instance-level control. Our UniFusion block enables instance-level conditions for text-to-image models, the ScaleU block improves image fidelity, and our Multi-instance Sampler improves generations for multiple instances. InstanceDiffusion significantly surpasses specialized state-of-the-art models for each location condition. Notably, on the COCO dataset, we outperform previous state-of-the-art by 20.4% AP_{50}^{box} *for box inputs, and 25.4% IoU for mask inputs.*

1. Introduction

Image generation models [8, 9, 18, 22, 26, 27, 44, 46, 50, 53, 69] trained on web-scale data have made tremendous progress in the recent years. Notably, text conditioned diffusion models now produce high quality images that contain the free form concepts specified in the text [12, 22, 44, 50, 53, 54]. While text-based control is useful, it does not always allow for precise and intuitive control over the output image. Thus, many different forms of conditioning, *e.g.*, edges, normal maps, semantic layouts have been proposed for better control [3, 7, 14, 15, 17, 34, 40, 41, 64, 65]. These richer controls enable a broader range of applications for the generative models in design, data generation [16, 68] *etc*. In this work, we focus on precise control over the *instances* in terms of their location and attributes in the output image.

We propose and study instance-conditioned image generation whereby a user can specify *every* instance in terms of its location and an instance-level text prompt to generate an image. The location can be specified using either a bounding box, an instance mask, a single point or a scribble. Practically, this allows for a flexible input where some instance locations maybe specified more precisely using masks, and others less precisely using points. The per instance text prompts allow for fine-grained control over the instance's attributes such as color, texture, *etc*. Our proposed instance-conditioned generation is a generalization of settings studied in prior work [4, 34, 65] that consider only one location format and do not use per instance captions.

Our model presents several design choices that enable more precise yet flexible control for instances in the output image. Since locations can be specified in a variety of formats, we present a unified way to parameterize and fuse their information during the generation process. Our unified modeling is simpler than prior work that uses separate architectures and strategies to model different location formats. Moreover, the unified modeling of location formats allows the model to exploit the shared underlying structure of instance locations which improves performance.

Through comprehensive evaluations, our method InstanceDiffusion outperforms state-of-the-art models specialized for particular instance conditions. We achieve a 20.4% increase in $\text{AP}_{50}^{\text{box}}$ over GLIGEN [34] when evaluating with bounding box inputs on COCO [36] `val`. For mask-based inputs, we obtain a 25.4% boost in IoU compared to DenseDiffusion [28] and a 36.2% gain in $\text{AP}_{50}^{\text{mask}}$ over ControlNet [65]. As prior methods do not study point or scribble inputs for image generation, we introduce evaluation metrics for these settings. InstanceDiffusion also demonstrates superior ability to adhere to attributes specified by instance-level text prompts. We obtain a substantial 25.2 point gain in instance color accuracy and a 9.2 point improvement in texture accuracy compared to GLIGEN.

Contributions. (1) In this paper, we propose and study instance-conditioned image generation that allows flexible location and attribute specification for multiple instances. (2) We propose three key modeling choices that improve results – (i) *UniFusion* (§ 3.2), which projects various forms of instance-level conditions into the same feature space, and injects the instance-level layout and descriptions into the visual tokens; (ii) *ScaleU* (§ 3.3), which re-calibrates the main features and the low-frequency components within the skip connection features of UNet, enhancing the model's ability to precisely adhere to the specified layout conditions; (iii) *Multi-instance Sampler* (§ 3.4), which reduces information leakage and confusion between the conditions on multiple instances (text+layout). (3) A dataset with instance-level captions generated using pretrained models (§ 3.5) and a new set of evaluation benchmarks and metrics for measuring the performance of location grounded image generation (§ 4.1). (4) Our unified modeling of different location formats significantly improves results over prior work (§ 4.2). We also show that our findings can be applied to previous approaches and boost their performance.

2. Related Work

Image Diffusion Models [22, 52, 54] learn the process of text-to-image generation through iterative denoising steps initiated from an initial random noise map. Latent diffusion models (LDMs) [47, 58] perform the diffusion process in the latent space of a Variational AutoEncoder [30, 58], for computational efficiency, and encode the textual inputs as feature vectors from pretrained language models [42]. DALL-E 2 [44] synthesizes images using the image space of CLIP [42]. In contrast, Imagen [50] diffuses pixels directly, without the need for latent images.

Image Generation with Spatial Controls is a form of conditional image synthesis task [14, 15, 20, 24, 34, 37, 55, 57, 59, 60, 62, 65, 67, 69], which introduces spatial conditioning controls to guide the image generation process. *Make-a-Scene*, *SpaText* [4], *GLIGEN* [34], and *ControlNet* [65] add finer grained spatial control, such as semantic segmentation masks, to large pretrained diffusion models by allowing users to include additional images that explicitly define their desired image composition. *GLIGEN* [34] can also support controlled image generation using discrete conditions such as bounding boxes. MultiDiffusion [6], DenseDiffusion [28], Attend-and-Excite [10], ReCo [63], StructureDiffusion [13], Layout-Guidance [11], and BoxDiff [61] add location controls to diffusion models without fine-tuning the pretrained text-to-image models. **Discussions.** ControlNet and GLIGEN require training separate models for each type of controllable input, which increases the overall complexity of the system and not effectively capture interactions across various controllable inputs. Moreover, while ControlNet focuses solely on spatial conditions and GLIGEN employs *object category* as the text prompt, the lack of training the models with detailed instance-level prompts not only limits user control but also hinders the model from effectively leveraging instance descriptions.

3. Instance Diffusion

We study adding precise, versatile instance-level control for text-based image generation.

Problem definition. We aim to improve instance-level control in image generation by focusing on two conditioning inputs for each instance, namely, its location and a text caption describing the instance. More formally, we want to learn an image generation model

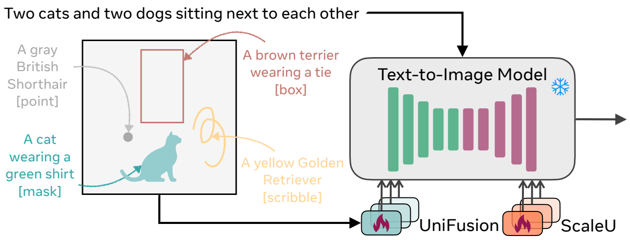

Figure 2. InstanceDiffusion enhances text-to-image models by providing additional instance-level control. In additon to a global text prompt, InstanceDiffusion allows for paired instance-level prompts and their locations to be specified when generating images. InstanceDiffusion is versatile, supporting a range of location forms, from the simplest points, boxes, and scribbles to more complex masks, and their flexible combinations.

$f(\mathbf{c}_g, \{(\mathbf{c}_1, \mathbf{l}_1), \ldots, (\mathbf{c}_n, \mathbf{l}_n)\})$ that is conditioned on a global text caption $\mathbf{c}_g$ and the per-instance conditions $(\mathbf{c}_i, \mathbf{l}_i)$ containing caption $\mathbf{c}_i$ and location $\mathbf{l}_i$ for n instances. This problem is similar to [4] and is a generalization of the 'open-set grounded text-to-image' [34] problem which does not consider per-instance captions. Our generalization allows for a generic and flexible way to control the scene-layout in terms of locations and attributes of the instances, as well as scene-level control via the global caption.

3.1. Approach overview

We introduce InstanceDiffusion (Figure 2) for instance-conditioned image generation. We support flexible ways to specify an object's location, *e.g.*, a single point, a scribble, a bounding box, and an instance mask. Since obtaining large-scale paired (text, image) data is much easier compared to (instance, image) data, we use a pretrained text-to-image UNet model that is frozen. We add our proposed learnable **UniFusion** blocks to handle the additional per-instance conditioning. UniFusion fuses the instance conditioning with the backbone and modulate its features to enable instance-conditioned image generation. In addition, we propose **ScaleU** blocks that improve the UNet's ability to respect instance-conditions by rescaling the skip-connection and backbone feature maps produced in the UNet. At inference, we propose **Multi-instance Sampler** which reduces information leakage across multiple instances.

Since obtaining a large paired (instance, image) dataset is difficult, we automatically generate a dataset with instance-level location and text captions using state-of-the-art recognition systems. Finally, we propose a new and comprehensive benchmark to evaluate the model's performance for instance-conditioned generation.

3.2. UniFusion block

The UniFusion block, illustrated in Figure 3, tokenizes the per-instance conditions $(\mathbf{c}_i, \mathbf{l}_i)$ and fuses them with the features, *i.e.*, visual tokens from the frozen text-to-image

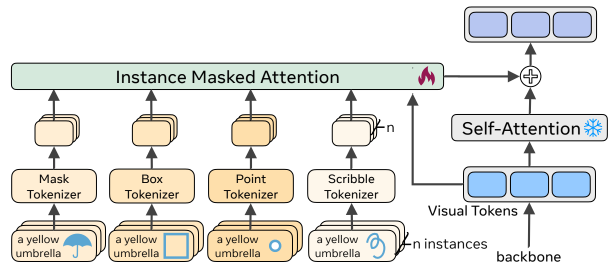

Figure 3. UniFusion projects various forms of instance-level conditions into the same feature space, seamlessly incorporating instance-level locations and text-prompts into the visual tokens from the diffusion backbone.

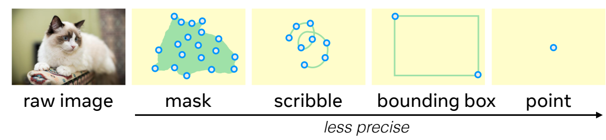

Figure 4. We represent different location condition formats as sets of points, with each format having varying quantities of points. Masks are represented as sparsely sampled points within the mask and uniformly sampled points from boundary polygons, bounding boxes by the top-right and bottom-right corners, and scribble are converted into uniformly sampled points.

model. Similar to [2, 34], the UniFusion block is added between the self-attention and cross-attention layers of the backbone. The per-instance location $\mathbf{l}_i$ can be specified in *one or more* location formats such as masks, boxes, *etc*. We now describe the key operations in the UniFusion block.

Location parameterization. As shown in Figure 4, we convert the four location formats - masks, boxes, scribbles, single point - into 2D points (denoted as $\mathbf{p}_i = \{(x_k, y_k)\}_{k=1}^{n}$ for instance i), with each 'format' having varying quantities of points n. A scribble is converted into a set of uniformly sampled points along the curve. We parameterize bounding boxes by their top-left and bottom-right corners. For *instance* masks, we convert them into a set of points sampled from within the mask and from boundary polygons.

Instance Tokenizer. We convert the 2D point coordinates $\mathbf{p}_i$ for each location using a Fourier mapping [56] $\gamma(\cdot)$ and encode the text prompt $\mathbf{c}_i$ using a CLIP text encoder $\tau_\theta(\cdot)$. Finally, we concatenate the location and text embeddings and feed them to an MLP to obtain a single token embedding $\mathbf{g}_i$ for the instance i: $\mathbf{g}_i = \text{MLP}([\tau_\theta(\mathbf{c}_i), \gamma(\mathbf{p}_i)])$. We use a different MLP for each location format. Moreover, the per-instance location $\mathbf{l}_i$ can be specified in one or more location formats. Thus, for each instance i, we obtain $\mathbf{g}_i^{\text{mask}}$, $\mathbf{g}_i^{\text{scribble}}$, $\mathbf{g}_i^{\text{box}}$, and $\mathbf{g}_i^{\text{point}}$. If an instance location is specified only using one format, *e.g.*, a single point, we use a learnable null token $\mathbf{e}_i$ for the other location formats:

$$\mathbf{g}_i = \text{MLP}([\tau_\theta(\mathbf{c}_i), s \cdot \gamma(\mathbf{p}_i) + (1-s) \cdot \mathbf{e}_i]) \tag{1}$$

where $s \in [0,1]$ refers to the presence of a location format.

(Optional) To better align with instance mask conditions, we can optionally add extra tokens from binary *instance* masks (dimensions N×H×W, with N as the instance number). These masks are resized to 512×512, and ConvNeXt-tiny [39] is used to get a 16×16 feature map, which is then flattened into grounding tokens and concatenated with $\{\mathbf{g}_i^{\text{mask}}\}_{i=1}^n$. These additional mask tokens may offer a minor boost in quantitative performance, yet enhance the model's accuracy in respecting object boundaries.

Prior work resizes *semantic* masks [34, 65] into the diffusion latent space of size 64×64, subsequently adding them into UNet inputs as extra channels. *Instances from the same semantic class are represented by one mask.* However, we found that this design choice hurts the performance, particularly in cases with overlapping instances and small objects.

Instance-Masked Attention and Fusion Mechanism. We denote the instance condition tokens, $\mathbf{g}$, per location format for all n instances by $\mathbf{G}$, and the m visual tokens, $\mathbf{v}$, from the backbone as $\mathbf{V}$. We apply masked self-attention (SA) to the instance condition tokens and the backbone features

$$\widetilde{\mathbf{V}} = \text{SA}_{\text{mask}}([\mathbf{V}, \mathbf{G}^{\text{mask}}, \mathbf{G}^{\text{scribble}}, \mathbf{G}^{\text{box}}, \mathbf{G}^{\text{point}}]) \quad (2)$$

We consider two design choices, ablated in Table 5, for the location inputs in Eq 2: 1) 'Format aware' (default) described above models each location format independently via concatenation. 2) 'Joint format' jointly models all location formats by concatenating embeddings from each format and converting them into a single embedding (via an MLP) to use in the masked self-attention.

We observed that vanilla self-attention, without masking, led to information leakage across instances, *e.g.*, color of one instance bleeding into another. Thus, we construct a mask $\mathbf{M}$ that prevents such leakage across instances:

$$\begin{aligned}\text{mask for } \mathbf{v}_k \cdot \mathbf{v}_j^T &: \mathbf{M}_{k,j} = -\inf \text{ if } I_{\mathbf{v}_k} \neq I_{\mathbf{v}_j} \\ \text{mask for } \mathbf{v}_k \cdot \mathbf{g}_i^T &: \mathbf{M}_{k,m+i} = -\inf \text{ if } I_{\mathbf{v}_k} \neq i\end{aligned} \quad (3)$$

where $I_{\mathbf{v}_k} = i$ if the visual token $\mathbf{v}_k$ falls within the region of the instance i defined by either a bounding box or an instance segmentation mask.

Finally, the output of the masked self-attention is added back to the backbone via gated addition

$$\mathbf{V} = \mathbf{V} + \tanh(\omega)\widetilde{\mathbf{V}}[:m] \quad (4)$$

where ω is a learnable parameter, initialized to 0, that controls the conditioning contribution of UniFusion.

3.3. ScaleU block

In the UNet model, each block merges the main feature map $\mathbf{F}_b$ with the lateral skip-connection features $\mathbf{F}_s$, passing the concatenated feature to the subsequent UNet block. FreeU [51] finds that the main backbone of UNet is critical for denoising, whereas its skip connections primarily contribute high-frequency features to the decoder. Concatenating these two features directly leads to the network neglecting the semantic content of the main features [51]. Therefore, FreeU suggests reducing the low-frequency components of the skip features and enhancing the main features using *channel-independent* and *empirically-tuned* values.

Our findings, however, demonstrate that for instance-conditioned image generation, a notable improvement can be achieved by using *channel-wise* and *learnable* vectors to dynamically re-calibrate $\mathbf{F}_b$ and $\mathbf{F}_s$. We introduce ScaleU, that has two *learnable*, *channel-wise* scaling vectors: $\mathbf{s}_b$, $\mathbf{s}_s$ for the main and skip-connected features, respectively. The main features $\mathbf{F}_b$ are scaled by a simple channel-wise multiplication: $\boldsymbol{F}_b' = \mathbf{F}_b \otimes (\tanh(\mathbf{s}_b) + 1)$. For the skip-connection features, we select the low-frequency (less than r_{thresh}) components using a frequency mask $\boldsymbol{\alpha}$ and scale them in the Fourier domain: $\mathbf{F}_s' = \text{IFFT}(\text{FFT}(\mathbf{F}_s) \odot \boldsymbol{\alpha})$. Here FFT$(\cdot)$ and IFFT$(\cdot)$ denote the Fast-Fourier and Inverse-Fast-Fourier transforms, $\odot$ is element-wise multiplication, and $\boldsymbol{\alpha}(r) = \tanh(\mathbf{s}_s) + 1$ if $r < r_{\text{thresh}}$ otherwise $= 1$, where r and r_{thresh} refers to the radius and threshold frequency, respectively. $\mathbf{s}_b$ and $\mathbf{s}_s$ are initially set to zeros.

Lightweight in parameters. The ScaleU module is incorporated into each of UNet's decoder blocks. It leads to a negligible ($< 0.01\%$) overall increase in the number of parameters and brings noticeable performance gains.

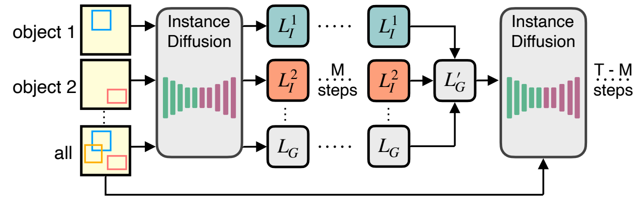

Figure 5. Model inference with **Multi-instance Sampler** to minimize information leakage across multiple instance conditionings.

3.4. Multi-instance Sampler

To further minimize the information leakage across multiple instance conditionings, we optionally use Multi-instance Sampler strategy during the model inference which improves the quality and fidelity of the generated image.

Specifically, Multi-instance Sampler (*cf*. Figure 5) involves: 1) For each of the n instances, run a separate denoising operation for M steps (less than 10% of the overall steps) to get the instance latents L_I. Note that, since our model is trained to generate an object within the location token specified for that object, we don't need to explicitly require the model to update the latent representation within the location. 2) Integrate the denoised instance latents $\{L_I^1, \cdots, L_I^n\}$ obtained from step (1) for each of the n objects with the global latent L_G, which is derived from all instance tokens and text prompts, by averaging these latents together. 3) Proceed to denoise the aggregated latent from step (2), utilizing all instance tokens and text prompts.

3.5. Data with Instance Captions

Standard object detection datasets [36] only contain a sparse category label, rather than a detailed caption, per object location. To capture more detailed information about instances and even instance parts, *e.g.*, attributes, we construct a dataset by using multiple models: ***1) Image-level label generation***: We employ RAM [66], a robust open-vocabulary image tagging model, to generate a list of common image-level tags. ***2) Bounding-box and mask generation***: We then use Grounded-SAM [31, 38] to produce bounding boxes and masks corresponding to these tags. These tags can at the instance-level, *e.g.*, a parrot, or at the part-level, *e.g.*, a bird's beak. ***3) Instance-level text prompt generation***: To generate instance-level text prompts that include descriptions of the instances, we crop the instances using their corresponding bounding boxes and create captions for these cropped instances using a pretrained Vision-Language Model (VLM) BLIP-V2 [32].

3.6. Implementation Details

We describe salient implementation details and provide the full details in the supplement.

Model training. We follow the same settings as GLIGEN [34] and initialize our model with a pretrained text-to-image model whose layers are frozen. We train the model with a batch size of 512 for 100K steps using the Adam optimizer [29] with a learning rate that is warmed up to 0.0001 after 5000 steps. More details are in appendix materials.

Training data. We automatically generate instance-level masks, boxes and captions following § 3.5. We obtain scribble by randomly sampling points within the masks. For single-points, we randomly select a point within a circular region of radius $0.1 \cdot r$, centered at the bounding box's center, where r is the length of the shortest side of the box.

4. Experiments

4.1. Experimental setup

Training data. Prior work, notably GLIGEN [34], relies on automatic annotations that use open-vocabulary detection models. These do not yield per-instance captions and different location formats such as scribble *etc*. (Note: 'mask' conditioning in prior work [4, 34] is per-category and not per-instance). Thus, to support the richer conditioning proposed in our work, we rely on recognition models as described in §§ 3.5 and 3.6 to generate instance-level annotations include different location formats (masks, boxes, scribbles, single-points) and per-instance captions. To ensure fair comparison to prior work [34], we use approximately the same number of images (5M) from an internal licensed dataset of natural images and paired global text.

Test data. We use standard benchmarks with bounding box and instance masks: 1) COCO [36] `val` with 80 classes; 2) large vocabulary instance segmentation dataset LVIS [19] `val` with over 1200 classes; 3) 250 selected samples ($\sim$2 objects per image) from COCO `val` as in [28]. We do not use the real images from the dataset, and only use the text and location conditions. Notably, we also do not use any information from the `train` splits of the data which makes our evaluations zero-shot.

Evaluation metrics for alignment to instance locations. We measure how well the objects in the generated image adhere to different location formats in the input.

Bounding box. We follow prior work [25, 28, 34, 45] and use the YOLO score. Specifically, we use a pretrained YOLOv8m-Det [25] detection model. We compare the model's detected bounding boxes on the generated image with the bounding boxes specified in the input using COCO's official evaluation metrics (AP and AR). We report $\mathrm{AP}_\mathrm{l}^\mathrm{box}$, $\mathrm{AP}_\mathrm{m}^\mathrm{box}$, and $\mathrm{AP}_\mathrm{s}^\mathrm{box}$, which evaluate the model's performance based on different object sizes.

Instance mask. We compare YOLOv8m-Seg [25]'s detected instance masks in the generated image to the masks specified in the input using the COCO AP and AR metrics. To compare with [28], we report the IOU score for the mask.

Scribble. Since prior work has not reported on alignment performance using scribble, we introduced a new evaluation metric using YOLOv8m-Seg. We report "Points in Mask" (**PiM**), which measures how many of randomly sampled points in the input scribble lie within the detected mask.

Single-point. Similar to scribble, the instance-level accuracy **PiM** is 1 if the input point is within the detected mask, and 0 otherwise. We then calculate the averaged **PiM** score.

Evaluation metrics for alignment to instance prompts:

Compositional attribute binding. We measure if the generated instances adhere to the attribute (color and texture) specified in the instance prompts. We use YOLOv8-Det to detect the bounding boxes. We feed the cropped box to the CLIP model to predict its attribute (colors and textures), and measure the accuracy of the prediction with respect to the attribute specified in the instance prompt. We use 8 common colors, *i.e.*, "black", "white", "red", "green", "yellow", "blue", "pink", "purple", and 8 common textures, *i.e.*, "rubber", "fluffy", "metallic", "wooden", "plastic", "fabric", "leather" and "glass".

Instance text-to-image alignment: We report the CLIP-Score on cropped object images (Local CLIP-score [4, 42]), which measures the distance between the instance text prompt's features and the cropped object images.

Global text-to-image alignment: CLIP-Score [42, 48] between the input text prompt and the generated image.

Human evaluation: We evaluate both the fidelity wrt

Location format input →	Boxes				Instance Masks					Points		Scribble	
Method	AP^{box}	AP^{box}_{50}	AR^{box}	FID (↓)	IoU	AP^{mask}	AP^{mask}_{50}	AR^{mask}	FID (↓)	PiM	FID (↓)	PiM	FID (↓)
Upper bound (real images)	50.2	66.7	61.0	-	-	40.8	63.5	58.0	-	-	-	-	-
GLIGEN [34]	19.6	35.0	30.7	27.0	-	-	-	-	-	-	-	$30.2^{\dagger}$	$32.4^{\dagger}$
ControlNet [65]‡	-	-	-	-	-	6.5	13.8	12.9	-	-	-	-	-
DenseDiffusion [28]	-	-	-	-	35.0 / 48.6	-	-	-	-	-	-	-	-
SpaText [4]‡	-	-	-	-	-	5.3	12.1	10.7	-	-	-	-	-
InstanceDiffusion	**38.8**	**55.4**	**52.9**	**23.9**	**61.6 / 71.4**	**27.1**	**50.0**	**38.1**	**25.5**	**81.1**	**27.5**	**72.4**	**27.3**
vs. prev. SoTA	**+19.2**	**+20.4**	**+21.8**	**-3.1**	**+25.4 / +22.8**	**+20.6**	**+36.2**	**+25.2**	-	-	-	**+42.2**	**-4.9**
InstanceDiffusion (hybrid)	**44.6**	**59.6**	**58.8**	**25.5**	-	-	-	-	-	**86.0**	**25.5**	**82.9**	**26.4**

Table 1. Evaluating different location formats as input when generating images. We measure the YOLO recognition performance (AP, AR) for the generated image wrt the location condition provided as inputs, and FID on the COCO `val` set. Most prior methods only support a handful of the location conditions. We observe that InstanceDiffusion, while using the same model parameters, supports various location inputs. In each setting, InstanceDiffusion substantially outperforms prior work on all metrics. †: GLIGEN's scribble-based results are derived by using the top-right and bottom-left corners as the bounding box for the region encompassed by the scribble. We measure the IoU using [28]'s official evaluation codes (left), and YOLOv8-Seg (right). ‡: ControlNet [65] (and SpaText [4]) only supports *semantic* segmentation mask inputs, and do not differentiate between instances of the same class. We assess ControlNet's AP^{mask} using its official mask conditioned Image2Image generation pipeline. Hybrid: we add instance masks as additional conditions.

Figure 6. Qualitative comparison of InstanceDiffusion *vs.* GLIGEN conditioned on multiple instance boxes and prompts. Prior work (bottom row) fails to accurately reflect specific instance attributes, *e.g.*, colors for the flower and puppies on the left, and not depicting a waterfall on the right. The generations also do not capture the correct instances, and are prone to information leakage across the instance prompts, *e.g.*, generating two similar instances on the right. InstanceDiffusion effectively mitigates these issues.

instance-level conditions (locations and text prompts) and the overall aesthetic of the generated images. We prompt users to select results that more closely adhere to the provided layout conditions and the accompanying instance captions. This evaluation is conducted on 250 samples, each accompanied by instance-level captions and bounding boxes.

4.2. Comparison with prior work

Single location format at inference. We assess the efficacy of multiple methods in generating images under diverse location formats and report results in Table 1. Since our evaluation uses recognition model (YOLO), we establish an upper bound by measuring the recognition performance on the real dataset images corresponding to the text and location conditions. Overall, our results show that InstanceDiffusion outperforms all prior work across various location conditions when measured across all evaluation metrics for object location and image quality. Next, we discuss the

Methods	Color		Texture		Human Eval
	Acc^{color}	$CLIP^{local}$	$Acc^{texture}$	$CLIP^{local}$	
GLIGEN	19.2	0.206	16.6	0.206	19.7
InstDiff	**54.4**	**0.250**	**26.8**	**0.225**	80.3
Δ	**+35.2**	**+0.044**	**+10.2**	**+0.019**	

Table 2. Attribute binding. We measure whether the attributes of the generated instances match the attributes specified in the instance captions. We observe that InstanceDiffusion outperforms prior work on both types of attributes. Human evaluators prefer our generations significantly more than the prior work.

results for each location format. Box input: InstanceDiffusion achieves the highest AP^{box} of 38.8 and AR^{box} of 52.9, outperforming the previous state-of-the-art by a significant margin, +19.2 and +21.8 for AP^{box} and AR^{box}, respectively. The reduction in FID score for InstanceDiffusion demonstrates its ability to produce high-quality images while adhering to the prescribed location conditions. Instance mask input imposes stricter constraints on the in-

Methods	AP	AP_{50}	AP_s	AP_m	AP_l	AP_r	AP_c	AP_f
Upper bound	44.6	57.7	33.2	55.0	66.1	31.4	44.5	50.5
GLIGEN [34]†	9.9	9.5	1.6	10.5	31.1	7.4	10.0	10.9
InstanceDiffusion	17.9	25.5	5.5	24.2	45.0	12.7	18.7	19.3
vs. prev. SoTA	**+8.0**	**+16.0**	**+3.9**	**13.7**	**+13.9**	**+5.3**	**+8.7**	**+8.4**

Table 3. Box inputs on LVIS `val`. We evaluate using a pretrained detector (ViTDet-L [33]) and obtain the upper bound by evaluating the detector on real images resized to 512×512. InstanceDiffusion significantly outperforms prior work across all metrics including object sizes, and class frequencies. †: reproduced results.

point	box	mask	PiM	AP^{box}	AP^{box}_{50}	AP^{mask}	AP^{mask}_{50}
✓	✗	✗	81.1	-	-	-	-
✓	✓	✗	85.6	38.8	55.4	-	-
✓	✓	✓	**86.0**	**44.6**	**59.6**	**27.1**	**50.0**

Table 4. Multiple location formats at inference improves performance and helps the model to better respect location conditions.

stance location than box input and is more challenging than the semantic masks studied in prior work [34, 65] that do not distinguish individual instances. Even in this challenging setting, InstanceDiffusion outperforms prior SOTA [28] significantly. Points and Scribble: Given the lack of prior studies that present quantifiable results for these location inputs, we introduce these novel evaluation metrics and benchmarks, establishing a new baseline for future research endeavors. Note that the term 'scribble' in ControlNet [65] refers to object boundary sketches rather than scribbles used in our work which follows [1, 5, 35].

Attribute binding. In Table 2, we measure whether the attributes (color and texture) of the generated instances match the attributes specified in the instance captions. We observe that attribute binding is challenging for the prior SOTA method, GLIGEN while InstanceDiffusion significantly improves on both color and texture binding. Adhering to texture seems to be more challenging than colors, *e.g.*, `wooden dog` *vs.* `red dog`, as reflected by the lower accuracies for all methods on this task. We compare the generations produced by both models using human evaluators and find that humans strongly prefer our generations over prior work (80.3% preference) confirming their high generation quality and controllability.

Challenging box inputs. In Table 3, we evaluate zero-shot performance on the challenging LVIS [19] dataset which has 15× more classes than COCO, and many more instances per sample (∼12 objects per images). InstanceDiffusion outperforms prior work across all metrics, and the gain is particularly strong on medium to large sized objects.

Multiple location formats at inference are analyzed in Table 4. We observe that using all formats together provides the best performance and more precise control on the instance location. This confirms the benefit of our design choice to model all location formats.

Qualitative results. Figure 6 provides qualitative comparisons between InstanceDiffusion and the previous SOTA method, GLIGEN [34], when given multiple instance boxes and associated text prompts. We see that GLIGEN often misinterprets specific instance attributes; *e.g.*, it incorrectly renders the colors of flowers and puppies on the left, and fails to produce a waterfall in the right images. GLIGEN also shows 'information leakage' across instance prompts (generating duplicate birds for the two images on the right). In Figure 7, we show more qualitative results using different location conditions for InstanceDiffusion.

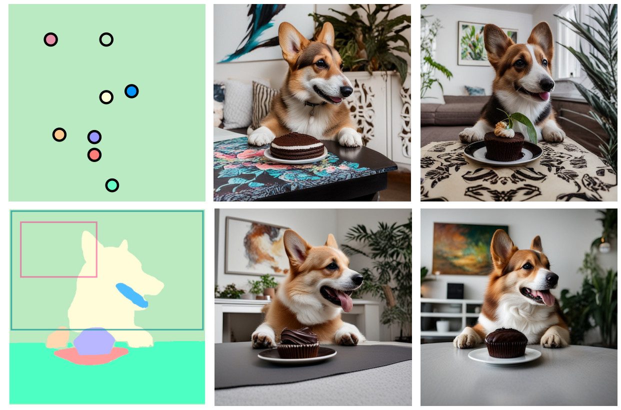

Image Caption: Cute Corgi at table in a living room with plants and painting on the wall. A chocolate cake is on the table. ***Instance Captions:*** 1) a Corgi sitting in front of a cupcake 2) Corgi's mouth and tongue 3) a plate 4) a chocolate cupcake on a plate 5) a white paw 6) a table 7) a living room with plants 8) oil painting on the wall

Figure 7. InstanceDiffusion image generation using various location conditions: points (row 1) and boxes+masks (row 2).

#	*FA Fusion*	*MaskAttn*	*ScaleU*	*Inst. Cap.*	*MIS*	AP^{mask}_{50}	Acc^{color}	FID (↓)
1	✓	✓	✓	✓	✓	50.0	55.4	25.5
2	✗	✓	✓	✓	✓	45.5**(5.5)**	49.4**(6.0)**	25.8**(0.3)**
3	✓	✗	✓	✓	✓	49.3**(0.7)**	53.1**(2.3)**	25.7**(0.2)**
4	✓	✓	✗	✓	✓	47.7**(2.3)**	52.2**(3.2)**	25.7**(0.2)**
5	✓	✓	✓	✗	✓	47.8**(2.2)**	38.2**(17.2)**	25.6**(0.1)**
6	✓	✓	✓	✓	✗	49.8**(0.2)**	49.5**(5.9)**	28.6**(3.1)**

Table 5. Contribution of each component evaluated by removing or adding it and measuring the impact of the generated image in terms of its instance location performance (AP), and instance attribute binding (Acc), and overall image quality (FID). When Format Aware (FA) fusion mechanism is disabled, we use the Joint format fusion mechanism instead. Top row is the default setting for InstanceDiffusion in the paper and we report the drop in performance for each subsequent row in **red**.

4.3. Ablation study

We ablate the components in InstanceDiffusion and use the COCO `val` set and provide mask, box and point location formats per-instance as input by default. **Some design choices** used in our method are ablated in in Table 6. We compare our proposed ScaleU block with FreeU in Table 6a. ScaleU leads to an improved localization AP suggesting that our learnable scaling of the backbone features outperforms the manually tuned FreeU. The impact of us-

versions →	FreeU [51]	ScaleU
AP_{50}^{box}	52.2	55.4

(a) ScaleU

methods →	w/o extra tokens	w/ extra tokens
AP_{50}^{mask}	46.7	50.0

(b) extra tokens from binary masks

format →	polygons	+inside
AP_{50}^{mask}	47.5	50.0

(c) mask parameterization

# points →	64	128	256	512
AP_{50}^{mask}	45.7	48.5	50.0	50.0

(d) # points per mask

Table 6. Ablating design choices where the default settings are indicated in gray. **(a)** Compared to FreeU, our proposed ScaleU block improves the models ability to respect location conditions. **(b)** Using extra tokens from binary instances masks can improve the mask AP. **(c)** Parameterizing the instance masks using points on their boundaries and inside is beneficial. **(d)** Increasing the number of points used to parameterize masks improves performance.

% of Steps →	0%	10%	20%	30%	36%	40%	50%
FID	28.6	27.8	27.4	25.8	25.5	25.0	27.0
AP_{50}^{mask}	49.8	49.8	49.4	49.4	50.0	49.2	48.3

Table 7. Multi-instance Sampler (MIS) lowers the FID and improves overall image quality. Location conditions: instance masks.

	GLIGEN [34]	w/ MIS	InstanceDiffusion	w/ MIS
Acc^{color}	19.2	**29.7**	49.5	**55.4**

Table 8. Multi-instance Sampler can be adapted for previous location conditioned work, yielding notable performance gains.

ing extra tokens generated from binary instance masks is explored in Table 6b. Lastly, for mask-conditioned input, Tabs. 6c and 6d show that points derived from both polygons and instance masks and using 128 points per instance mask gives the optimal performance.

Contribution of each component its effect on image generation is measured in Table 5. We compare using different design choices for the fusion mechanism in UniFusion that fuse the location condition embeddings with the backbone text-to-image features: Format Aware fusion (row 1) or the Joint Format fusion (row 2). We find that making the fusion mechanism format-aware significantly improves performance since the location formats specify varying degrees of control on the instance location. Comparing rows 1, 3 shows that using Instance-Masked Attention for fusing the location features helps the model focus on instance-specific regions and thus improves attribute binding (color accuracy). Removing ScaleU (rows 1, 4) causes a significant drop in AP_{50}^{mask} and Acc^{color} scores. This underscores the importance of dynamically adjusting the channel weights of both skip connected and backbone features. In row 5, we observe that our generated instance captions are critical for learning attribute binding, as indicated by the 17% drop in Acc^{color} after removing them. Finally, row 6 shows that Multi-instance Sampler (MIS) improves the overall image quality (lower FID) and attribute binding (color accuracy).

Multi-instance Sampler The impact of the proportion of MIS steps used in inference is explored inn Table 7. MIS can effectively improve the quality of the generated images and attribute binding when the MIS percentage is below 36%. As in Table 8, we apply Multi-instance Sampler to other location-conditioned text-to-image models and observed significant gains for the attribute binding ability. These results confirm that MIS minimizes information leakage and that it can be easily used to improve other methods.

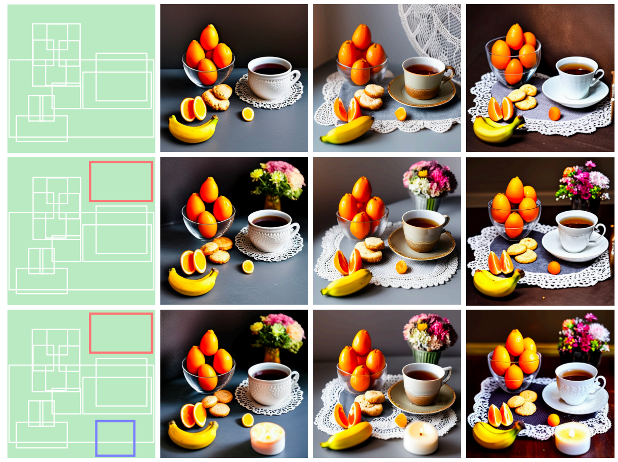

Image Caption: A cup of tea with tangerines, bananas, and cookies on the table. high quality. professional photo.
Instance Captions: 1) a cup of tea on a lace doily 2) a close up of three oranges on a black background 3) oranges in a glass bowl on a table 4) a tray of pastries on a table with oranges 5) a close up of some cookies on a table 6) oranges in a glass bowl 7) oranges in a glass bowl 8) an orange that has been cut in half on a table 9) an orange is cut in half 10) bananas 11) a bouquet of flowers on a table 12) a bouquet of flowers on a table 13) A candle

Figure 8. InstanceDiffusion can also support **iterative image generation**. Using the identical initial noise and image caption, InstanceDiffusion can progressively add new instances (like a bouquet of flowers in row two and a candle in row three), while minimally altering the pre-generated instances (row one). More results on iterative image generation that supports instance editing, replacing, moving and resizing can be found in appendix materials.

Application: Iterative generation. Since InstanceDiffusion allows for precise control over the instances, we show a useful application that benefits from this property in Figure 8. InstanceDiffusion allows users to selectively insert objects into precise locations while preserving the integrity of previously generated objects and the global scene. We hope that the precise control enabled by InstanceDiffusion will lead to many other such useful applications.

5. Conclusions, Limitations and Future Work

We presented InstanceDiffusion which enables precise instance-level control for text-to-image generation and significantly outperforms all prior work in terms of complying with instance attributes and accommodates a variety of location formats – masks, boxes, scribbles and points. Our studies indicate that there is a noticeable disparity in the generation quality of small objects compared to larger ones. We also find that texture binding for instances poses a challenge across all methods tested, including InstanceDiffusion. Improving instance conditioning for these cases is an important direction for future research.

References

[1] Eirikur Agustsson, Jasper RR Uijlings, and Vittorio Ferrari. Interactive full image segmentation by considering all regions jointly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11622–11631, 2019. 7

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 3

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 1

[4] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *ICCV*, 2023. 2, 3, 5, 6

[5] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 392–399, 2014. 7

[6] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023. 2

[7] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 1

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1

[9] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 1

[10] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023. 2

[11] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023. 2

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1

[13] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022. 2

[14] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023. 1, 2

[15] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 1, 2

[16] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Laurent Itti, and Vibhav Vineet. Dall-e for detection: Language-driven context image synthesis for object detection. *arXiv preprint arXiv:2206.09592*, 2022. 1

[17] Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Nicu Sebe, Trevor Darrell, Zhangyang Wang, and Humphrey Shi. Pair-diffusion: Object-level image editing with structure-and-appearance paired diffusion models. *arXiv preprint arXiv:2303.17546*, 2023. 1

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[19] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 5, 7

[20] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 14

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 12

[23] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 13

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[25] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 2023. 5

[26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[28] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *ICCV*, 2023. 2, 5, 6, 7

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 14

[30] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2, 12

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 5

[32] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 5

[33] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. 7

[34] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 12, 14

[35] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016. 7

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5

[37] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017. 2

[38] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 5

[39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 4

[40] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1

[41] Dong Huk Park, Grace Luo, Clayton Toste, Samaneh Azadi, Xihui Liu, Maka Karalashvili, Anna Rohrbach, and Trevor Darrell. Shape-guided diffusion with inside-outside attention. *arXiv preprint arXiv:2212.00210*, 2022. 1

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 5, 12

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 12

[44] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 5

[46] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 1

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2, 12

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 12

[50] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2

[51] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *arXiv preprint arXiv:2309.11497*, 2023. 4, 8, 12

[52] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2, 12

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 12

[54] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2, 12

[55] David Stap, Maurits Bleeker, Sarah Ibrahimi, and Maartje Ter Hoeve. Conditional image generation and manipulation for user-specified content. *arXiv preprint arXiv:2005.04909*, 2020. 2

[56] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 3, 12

[57] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 2

[58] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[59] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 2

[60] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2

[61] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023. 2

[62] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 2

[63] Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255, 2023. 2

[64] Lisai Zhang, Qingcai Chen, Baotian Hu, and Shuoran Jiang. Text-guided neural image inpainting. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1302–1310, 2020. 1

[65] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2, 4, 6, 7, 12

[66] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023. 5

[67] Zhu Zhang, Jianxin Ma, Chang Zhou, Rui Men, Zhikang Li, Ming Ding, Jie Tang, Jingren Zhou, and Hongxia Yang. Ufc-bert: Unifying multi-modal controls for conditional image synthesis. *Advances in Neural Information Processing Systems*, 34:27196–27208, 2021. 2

[68] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, Weiming Zhang, and Nenghai Yu. X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. In *International Conference on Machine Learning*, 2023. 1

[69] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1, 2